**ACL** Access Control List

格式：**[scheme:id:permissions]**

scheme：采用哪种权限机制

id：允许访问的用户

permissions：权限组合

**scheme ---**

world：只有一个用户anyone，world:anyone:[permissions]

auth：认证登录，需要注册用户且有相应权限，auth:user:password:[permissions]

**digest**：需要密码加密，digest:username:BASE64(SHA1(password)):[permissions]

ip：限制ip进行访问，ip:192.168.1.1:[permissions]

super 超级管理员，文档配置

**permissions --- crdwa**

create 创建子节点

read 获取节点/子节点

write 设置节点数据

delete 删除子节点

admin 设置权限

用户登录 **addauth digest username:password**

**Four Letter Words**

echo [command] | nc [ip] [port]

[**stat**] 查看zk状态信息

[**ruok**] 查看zk是否启动

[**dump**] 列出会话Session和临时节点ephemeral

[**conf**] 查看服务器配置

[**cons**] 连接到服务器的客户端信息

[envi] 环境变量

[mntr] 监控zk的健康信息

[wchs] 查看watch信息

[wchc] [wchp] 查看watch和session，watch和path

**集群**

创建集群，在各自的zookeeper配置文件中，添加相同的配置

server.1=192.168.1.100:2888:3888

server.2=192.168.1.100:2889:3889

server.3=192.168.1.100:2890:3890

在各自的dataDir下创建myid文件，内容分别为1 2 3

启动Server

zookeeper的数据一致性靠Paxos算法保证，其思路是：由leader发起提案，并为该提案有一个编号，各follower拿到提案，和自己手里的编号比较，如果大于自己的编号，就进行投票；只有获取到半数以上的票才能通过提案，通过后各自更新数据和编号。

**zk原生api的不足之处**

超时重连，不支持自动，需要手动操作；

watch注册一次后会失效；

不支持递归创建节点

解决方案 **Apache curator**

使用NodeCache，监听数据节点的变更，解决watch只能监听一次的问题

nodeCache.getListenable().addListener(new NodeCacheListener(){

public void nodeChanged() throws Exception {

}

});

监听子节点，使用PathChildrenCache

**zk-watcher统一更新N台节点的配置文件**，思路：让N台节点都监听zk服务器上同一个节点，当节点数据发生变化时，N台服务器会进入到相应的回调事件中，解析节点数据并做出相应的增删改动作。